Multi-Path Selection for Multiple Description Encoded Video Streaming

Ali C. Begen and Yucel Altunbasak
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250
Email: {acbegen, yucel}@ece.gatech.edu

Ozlem Ergun
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0205
Email: oergun@isye.gatech.edu

Abstract— **This paper presents a new framework for multimedia streaming that integrates the application and network layer functionalities to meet such stringent application requirements as delay and loss. The coordination between these two layers provides more robust media transmission even under severe network conditions. In this framework, a multiple description source coder is used to produce multiple independently-decodable streams that are routed over partially link-disjoint (non-shared) paths to combat bursty packet losses. We model multi-path streaming and propose a multi-path selection method that chooses a set of paths maximizing the overall quality at the client. Overlay infrastructure is then used to achieve multi-path routing over these selected paths. The simulation results show that the average peak signal-to-noise ratio (PSNR) improves by up to 8.1 dB, if the same source video is routed over intelligently selected multiple paths instead of the shortest path or maximally link-disjoint paths. In addition to PSNR improvement in quality, the end-user experiences a more continual streaming quality.**

I. Introduction

With the advances in audio-visual encoding standards and broadband access networks, multimedia communications (MMC) has become closer to reality. The popularity of the Internet further stimulated the demand for MMC services. However, the underlying infrastructure, *i.e.*, routing and transport protocols, and control and signalling mechanisms, has not been sufficiently developed to support such challenging applications. Therefore, early efforts largely failed, and this experience encouraged researchers to seek new techniques to support delay-sensitive multimedia applications over lossy networks.

In this paper, we propose a novel framework for multimedia streaming over packet networks. While generating several streams with a multiple description (MD) source coder, we transmit these streams over diverse paths by using an overlay infrastructure. Current routing algorithms are mainly destination-based and they use the hop counts as their distance metric. Although, this idea performs well for data applications, it generally underperforms for real-time multimedia applications. One of the key mechanisms needed for incorporating MDC into routing algorithms is the ability to efficiently select multiple paths between the server and client. An easy solution would be to use totally link-disjoint paths. However, these paths or the shortest paths do not necessarily make the best use of error-resiliency properties of MDC. Moreover, totally link-disjoint paths are rarely available between two end hosts. *The routing algorithms for MDC-enabled multimedia streaming system should choose paths in such a way that the overall streaming quality is maximized or equivalently, the overall distortion is minimized.*

The contemporaneous work about using MDC with path diversity is [1]. Based on this work, Apostolopoulos *et al.* discuss MD streaming within content delivery networks in [2]. The authors study video streaming using different reference picture selection schemes and path diversity in [3], [4] and show that the quality can be increased in doing so. Also, the study [5] investigates the performance of five different traffic dispersion techniques for high bandwidth MPEG-2 streaming data. All of these research studies propose methods to increase the reliability of the streamed video. However, to emulate multi-path transport they assume the availability of multiple statically independent paths between the end hosts, which is not feasible in many cases. Furthermore, they neither provide a particular network model nor a path selection method. In this study, we address these open problems and propose a method to select optimal multiple paths as a function of the network parameters, media characteristics and application requirements.[1]

The rest of the paper is organized as follows. In Section II, we give an overview of MDC. Section III introduces the envisioned network model in this work. Multi-path selection is discussed in Section IV. Simulation results are presented in Section V. Finally, Section VI concludes the paper.

II. Multiple Description Coding Theory

Multiple Description Coding (MDC) is a source coding that generates several descriptions such that different levels of reconstruction qualities can be obtained from different subsets of these descriptions. In contrast to Layered Coding, there is no hierarchy among the descriptions so that each description may be independently decoded. This property makes MDC highly suitable for packet networks where no prioritization

[1]It should be mentioned that the proposed multi-path selection method is only *optimal* within the context of the network models and objective functions that are considered in this paper.

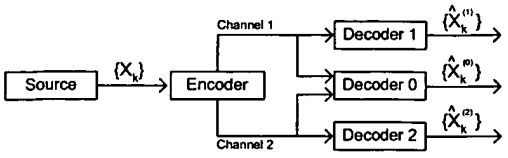

Fig. 1. A typical multiple description codec with two descriptions.

exists among the packets. MDC provides this robustness at the expense of a small reduction in the compression efficiency.

A typical MDC system is shown in Fig. 1. A sequence of source symbols $\{X_k\}_{k=1}^{N}$ is input to the multiple description (MD) encoder. The MD encoder produces two descriptions, which are then transmitted over possibly different channels. If both descriptions are received intact at the client, then we use the *central decoder* to produce the highest quality signal. However, if only one description is received error-free, then, the corresponding *side decoder* is used. We represent the reconstructed signal at the i^{th} decoder by $\{\hat{X}_k^{(i)}\}_{k=1}^{N}$ and the distortion by D_i. R_i denotes the rate in terms of the number of bits per source sample on the i^{th} channel.

Our objective is to estimate the end-to-end video quality in terms of path parameters. We start our derivation by relating the rate of the descriptions to the reconstructed signal quality, or equivalently, the distortion. For the sake of simplicity, we consider only two descriptions generated at the source. We define the average distortion at the i^{th} decoder as

$$D_i = E\left\{\frac{1}{N}\sum_{k=1}^{N} d(X_k, \hat{X}_k^{(i)})\right\} \quad i = 0, 1 \text{ and } 2, \qquad (1)$$

where $d(.)$ can be any distortion measure. In this paper, we take $d(.)$ as the squared Euclidian norm.

To develop analytic expressions for the average distortion, we need to choose a media type and source model. In the rest of the paper, we assume that the source is a compressed video stream. However, note that methodology presented here applies to any source with its specific distortion model. That is, the proposed path selection method chooses the paths resulting in the maximal quality irrespective of the type of the streamed media and its distortion model.

In compressed-video applications, the distribution of Discrete Cosine Transform (DCT) coefficients can be modelled as Laplacian. For a Laplacian source with variance σ^2, the rate-distortion function is upper and lower bounded by [6]

$$\frac{e}{\pi}\sigma^2 2^{-2R_i} \leqslant D_{laplacian}(R_i) \leqslant \sigma^2 2^{-2R_i}. \qquad (2)$$

In Section IV, we will use (2) to derive a relation between the source rate (R_i) and streaming distortion.

There are various techniques for generating multiple descriptions. One of the most straightforward methods is time-domain partitioning. For instance, separating the even and odd numbered frames in a video sequence into two groups produces two descriptions. Likewise, the sequence may also be partitioned in the spatial-domain. Other popular approaches are Multiple Description Scalar Quantization (MDSQ) and Multiple Description Transform Coding (MDTC). In MDSQ, multiple quantizers are used to generate descriptions whereas correlating transforms are used in MDTC. For brevity, we will not go into further details of MDC. For an excellent survey, interested readers are referred to [7].

III. Envisioned End-to-end Network Model

In this study, we assume the availability of an overlay network that spans several nodes including the server and client similar to the Resilient Overlay Networks (RONs) and X-Bone [8], [9]. In such an overlay network, we are able to achieve MD streaming by application-layer routing. Although it is not as efficient as network-layer routing, for overlays, application-layer routing does not require any network support, hence, can easily be implemented in today's Internet. Here, the overlay nodes, called O-nodes, are logically connected to each other to build an overlay network that will be exploited to enable multi-path routing. With this aspect, we first find the optimal set of paths from the server to the client. Then the path information, *i.e.*, addresses of the O-nodes to be traversed, is encoded into the packet payload. This information is used by the O-nodes to forward the incoming packet to the next O-node on the path. Between the O-nodes, the packets are routed by the underlying routing mechanism such as the shortest path routing. Hence, the overall procedure is *transparent* to the physical network.

Besides enabling multi-path routing, each O-node continuously monitors the statistics of the flows between other O-nodes and itself. To discover the overlay topology and link characteristics, every O-node exchanges information among other O-nodes by a routing protocol. The throughput, packet loss rate, delay and jitter statistics are then used to compute application-specific routes. The details of implementing such protocols are further discussed in [8].

A. Definitions

Now, we will define the network and the link parameters. Recall that, we are operating on an overlay network. Hence, a *link* stands for a logical link.

Definition 1: A *network* is defined by the tuple (N, L) where $N = \{N_1, N_2, ..., N_n\}$ represents the nodes in the network and L is an $n \times n$ matrix. $L(u, v) = 1$ if a direct link, denoted by $l_{u,v}$, exists between nodes N_u and N_v and $L(u, v) = 0$, otherwise.

Definition 2: A *path* $\mathcal{P}_{S,C}$ is defined by a set of nodes and links connecting nodes N_S and N_C. The set of the nodes on this path is represented by $\mathcal{N}_{S,C} \subseteq N$ and the set of the links on this path is denoted by $\mathcal{L}_{S,C}$.

Let $\mathcal{P}_{S,C}^1$ and $\mathcal{P}_{S,C}^2$ be any two paths. These paths are said to be totally link-disjoint if and only if $\mathcal{L}_{S,C}^1$ and $\mathcal{L}_{S,C}^2$ do not have any common element. If the sets $\mathcal{L}_{S,C}^1$ and $\mathcal{L}_{S,C}^2$ intersect, the links belonging to both paths are referred as *joint links*, whereas the rest of the links are referred as the *disjoint links*.

B. Link Parameters

The parameters for the link $l_{u,v}$ are defined as follows:

- $b_{u,v}$ denotes the bandwidth between nodes N_u and N_v.
- $p_{u,v}$ denotes the packet loss probability between nodes N_u and N_v. $p_{u,v}$ values for different links are assumed to be independent.
- $t_{u,v}$ denotes the minimum delay between nodes N_u and N_v. This includes processing, transmission and propagation delays, but not varying queuing delays.
- $j_{u,v}$ denotes the average jitter between nodes N_u and N_v. This jitter is mainly due to the varying queuing delays. Together with $t_{u,v}$, $j_{u,v}$ determines the forward trip time.

IV. Multi-path Selection

In this section, we introduce the multi-path selection method step by step. Let's consider two paths, $\mathcal{P}^1_{S,C}$ and $\mathcal{P}^2_{S,C}$, between the nodes N_S and N_C, which are referred as a *path pair*. We do not impose that the paths should be different, *i.e.*, $\mathcal{P}^1_{S,C}$ may be the same as $\mathcal{P}^2_{S,C}$. Our goal is to estimate the end-user quality in terms of the link parameters making up these paths. In other words, we are looking for a function f that determines the average distortion at the client, $\overline{D}_c$, based on the link parameters. That is,

$$\overline{D}_c = f(b_{u,v}, p_{u,v}, t_{u,v}, j_{u,v}) \text{ for } \forall l_{u,v} \in \mathcal{L}^i_{S,C},\ i = 1, 2. \quad (3)$$

Recall that our ultimate goal is to find the path pair that minimizes the cost function, f. For two descriptions, we have four possible cases determined by the description on-time arrivals. Let $D_{0,0}$ ($D_{1,1}$) denote the distortion when both descriptions arrive intact on time (are lost or delayed). Similarly, let $D_{0,1}$ ($D_{1,0}$) denote the distortion when the first (second) description arrives intact on time, but the other one is lost or delayed. Given these distortions, we can write the average distortion at the client as

$$\begin{aligned}\overline{D}_c = & \overbrace{Prob\{\text{Both received on time}\}}^{P_{0,0}} \times D_{0,0} \\ & + \overbrace{Prob\{1^{\text{st}} \text{ received on time } \& \ 2^{\text{nd}} \text{ lost or delayed}\}}^{P_{0,1}} \times D_{0,1} \\ & + \underbrace{Prob\{1^{\text{st}} \text{ lost or delayed } \& \ 2^{\text{nd}} \text{ received on time}\}}_{P_{1,0}} \times D_{1,0} \\ & + \underbrace{Prob\{\text{Both lost or delayed}\}}_{P_{1,1}} \times D_{1,1},\end{aligned} \quad (4)$$

where the success probabilities, $P_{0,0} - P_{1,1}$, will be derived later in this section. The shared parameters due to the joint links play a critical role in the path pair selection process. Hence, the selection process should consider all possible path pairs. One cannot evaluate the paths individually, rank them, and then choose the best two. This does not necessarily give the optimal pair due to the dependencies between the paths. For the sake of simplicity, throughout the section we drop the subscripts and superscript and use i to denote the i^{th} path, where $i \in \{1, 2\}$. We start our derivations with distortion.

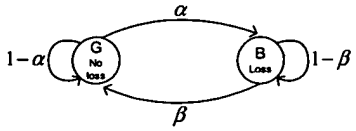

Fig. 2. State transition diagram for the Gilbert-Elliott model.

A. Bandwidth - Distortion Relation

Let B_i denote the end-to-end available bandwidth on the path $\mathcal{P}_i$. Clearly, B_i is the minimum bandwidth over all the links on $\mathcal{P}_i$. That is, $B_i = min\{b_{u,v} : \forall l_{u,v} \in \mathcal{L}_i\}$. In Section II, we have given the upper and lower bounds for the distortion of a Laplacian source. Now, we will give the relation between R_i in (2) and B_i. Assuming that the i^{th} description is encoded with a rate of R_i bits per pixel (bpp), and the source video has a resolution of $W \times H$ (pixels/frame) at a frame rate of F (frames/second), we have

$$R_i = \frac{B_i}{W \times H \times F \times c} \quad \text{(bpp)}, \quad (5)$$

where c is a known constant that depends on the chroma subsampling format. Typically, a streamed video is in 4:2:0 format and the corresponding c is 1.5. Since there is only a coefficient difference between the lower and upper bounds in (2), for our purposes, it suffices to give the distortion as a function of the aggregated bandwidth of the path pair (B) as

$$D(B) = \sigma^2 2^{-2 \times \frac{B}{W \times H \times F \times c}}. \quad (6)$$

Given this relation, we can compute the distortion terms in (4) by plugging the corresponding B values into (6). First, for $D_{0,0}$ the aggregated bandwidth is typically the sum of the individual bandwidths, *i.e.*, $B = B_1 + B_2$ provided that the joint links have available bandwidths higher than this sum. However, if this is not the case, B is then limited by the bottleneck joint-link capacity. Then, to compute $D_{0,1}$ and $D_{1,0}$, we take $B = B_1$ and $B = B_2$, respectively. Finally, $D_{1,1}$ is computed by letting $B = 0$.

B. Computation of Success Probabilities

The next step in the computation of (4) is to find the success probabilities ($P_{0,0} - P_{1,1}$). Each of these probabilities are comprised of two components. The first component computes the probability that the description arrives at the receiver in a finite duration, *i.e.*, it is not lost during the transmission. We will refer this component as *arrival probability*. On the other hand, for the second component we are interested in computing the probability that the description arrival occurs before a pre-specified deadline. This component will be referred as *on-time delivery probability*.

B.1. Arrival Probability

We adopted a well-known two-state Markovian Gilbert-Elliott (GE) model [10], [11] to describe the temporal behavior of packet losses on a link. The state transition diagram of the GE model is given in Fig. 2. In this figure, G (good) and B (bad) represent the packet arrival and loss states, respectively

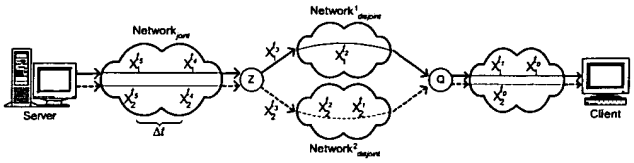

Fig. 3. Path segregation method not only reduces the topology complexity but also allows us to compute the arrival probabilities accurately.

and the corresponding steady-state probabilities are given by $\pi_G = \frac{\beta}{\alpha+\beta}$ and $\pi_B = \frac{\alpha}{\alpha+\beta}$. For the link $l_{u,v}$, we compute the transition probabilities by

$$\alpha_{u,v} = \frac{p_{u,v} \times \beta_{u,v}}{1 - p_{u,v}} \quad \text{and} \quad \beta_{u,v} = \frac{1}{L_{u,v}}, \tag{7}$$

where $L_{u,v}$ is the average burst length on this link. Typically, $L_{u,v}$ is a number between 4 and 7 for applications running over UDP. It is important to note that if temporal dependency of bursty packet losses was ignored, the optimal solution would be to send all descriptions over the best single path.

The GE model provides a simple expression for the characterization of individual links. However, derivation of the end-to-end arrival probabilities on partially link-disjoint paths is not trivial. Next, we focus on this crucial step.

B.1.a. Path Segregation

The GE model successfully captures bursty loss characteristics of a flow. However, when several flows are interleaved on the same link, the burst observed by a particular flow may become less significant. As enough number of foreign packets are introduced in between two consecutive packets of a flow, packet loss events become less and less correlated and eventually, the GE model reduces to a Bernoulli model.

Consider the topology depicted in Fig. 3 where two candidate paths split at node N_Z, and merge later at node N_Q. The joint links between the server and the segregation point (N_Z) form a sub-network denoted by Network_{joint}. Beyond the node N_Z, the disjoint links traversed by each path until the node N_Q are also grouped to form the corresponding sub-networks that are denoted by $\text{Network}^i_{disjoint}$.

Assume that the source generates two correlated descriptions, $X_1^{t_i}$ and $X_2^{t_i}$, with a period of Δt where t_i denotes the time at which the descriptions are generated. The value of Δt is chosen large enough so that the descriptions generated at the time instants t_i and $t_i + \Delta t$ will not fall into the same burst period, *i.e.*, the loss events for $X_1^{t_i}$ and $X_1^{t_i+\Delta t}$ will be independent of each other. Further assume that each description is one packet and transmitted as soon as it is ready. Consequently, the packets belonging to $X_1^{t_i}$ and $X_2^{t_i}$ will arrive at node N_Z approximately at the same time (back-to-back). However, once the paths split at node N_Z, these two packets will be routed over different paths. The chances are that these packets will not arrive at the node N_Q within a typical burst duration. Moreover, beyond the node N_Q, it is not likely that these packets will again meet in the same burst duration. From this observation, we deduce that once the paths split, the loss events on the rest of the paths should be considered independent despite a possible merge at a later node. This suggests us to employ a joint GE model for Network_{joint}, and individual Bernoulli models for $\text{Network}^i_{disjoint}$. We refer this method as *path segregation*.

We have simulated several topologies to check the validity of path segregation. We compared the packet loss burst size histograms generated by the original and approximated topologies. As expected as the delay difference between the paths became comparable to the average burst duration, the model error reduced almost to zero. In a real network, the disjoint paths will always have enough delay difference since they will experience uneven delay and jitter.

B.1.b. Link Aggregation

Generally, the joint sub-path in Network_{joint} is a combination of several Markovian links. Although it is straightforward to compute the arrival probabilities by considering each of these links individually, this computation will be greatly simplified if we can approximate the combination of these links by a single Markovian link. However, for the approximated link one has to find the joint GE parameters, namely α_{joint} and β_{joint}. Recall that consecutive links' packet loss rates are independent. Hence, we can directly compute the packet loss rate for Network_{joint} as

$$p_{joint} = 1 - \prod_{\forall l_{u,v} \in \text{Network}_{joint}} (1 - p_{u,v}). \tag{8}$$

However, estimating the joint average burst length, L_{joint} is not as trivial. This is because bursty periods in a stream accumulate as the stream goes over more links, *i.e.*, effective average burst length increases. Although there are several approaches to handle this problem [10], [11], they usually require higher order state analysis that renders the proposed approach impractical. For this reason, we estimate L_{joint} with the following heuristic:

$$L_{joint} = \frac{1}{p_{joint}} \times \sum_{\forall l_{u,v} \in \text{Network}_{joint}} p_{u,v} \times L_{u,v}. \tag{9}$$

Once we have the p_{joint} and L_{joint}, we compute α_{joint} and β_{joint} by plugging p_{joint} and L_{joint} into (7). This method is referred as *link aggregation* since it combines several Markovian links into a single one. The simulation results show that the heuristic in (9) successfully estimates L_{joint}.

On the other hand, since the disjoint sub-paths in $\text{Network}^i_{disjoint}$ are modelled by Bernoulli, we can directly convert these links into a single Bernoulli link whose loss probability is given by

$$p^i_{disjoint} = 1 - \prod_{\forall l_{u,v} \in \text{Network}^i_{disjoint}} (1 - p_{u,v}). \tag{10}$$

Finally, the arrival probability of a path is the product of the arrival probabilities on its joint and disjoint sub-paths.

B.2. On-time Delivery Probability

The second step in computing the success probabilities is to derive on-time delivery probabilities. Let T_i denote the minimum end-to-end delay for the path $\mathcal{P}_i$. We can write T_i as $T_i = \sum_{\forall l_{u,v} \in \mathcal{L}_i} t_{u,v}$. Obviously, we compel T_i to be smaller than the pre-specified deadline period. Otherwise, it means that all descriptions transmitted over this path will be obsolete by the time they are received.

Let J_i denote the average end-to-end jitter on path $\mathcal{P}_i$. We divide J_i into two components, j_{joint} and $j^i_{disjoint}$, as the jitters experienced in Network_{joint} and $\text{Network}^i_{disjoint}$, respectively. Assuming that jitters on the consecutive links are independent, we compute J_i as follows:

$$J_i = \underbrace{\sum_{\forall l_{u,v} \in \text{Network}_{joint}} j_{u,v}}_{j_{joint}} + \underbrace{\sum_{\forall l_{u,v} \in \text{Network}^i_{disjoint}} j_{u,v}}_{j^i_{disjoint}}. \quad (11)$$

Let $T_{deadline}$ represent the maximum tolerable delay for a target application. Generally, different paths have different end-to-end delays, and hence, different tolerances for jitter. We define t^i_{max} as the maximum jitter that can be tolerated on path $\mathcal{P}_i$. By definition, $t^i_{max} = T_{deadline} - T_i$. Then, it is easy to see that the on-time delivery probability can also be given as the probability that J_i is smaller than t^i_{max}.

Finally, we have all the information required to write success probabilities ($P_{0,0} - P_{1,1}$) in terms of path parameters. Recall that, in (4), $P_{0,0}$ computes the probability that both descriptions arrive at the client intact on time. To find this value, we first compute the arrival probability of transmitting both descriptions intact over Network_{joint} and individual arrival probabilities of transmitting each description intact over $\text{Network}^i_{disjoint}$. Since these events are independent, we multiply them to find the arrival probability of both descriptions. Secondly, we compute the probability that the descriptions will arrive at the client before the pre-specified deadline. Finally, we multiply this value with the arrival probability to get $P_{0,0}$.

On the other hand, $P_{0,1}$ considers the cases where the first description is received intact on time and the second description is (i) lost in Network_{joint} or in $\text{Network}^2_{disjoint}$ or (ii) delayed beyond the play-out time although transmitted intact. Hence, we separately compute the corresponding probabilities of these three events and then add them up to get $P_{0,1}$. $P_{1,0}$ can also be computed in the same manner. Lastly, $P_{1,1}$ is the complement of other given probabilities. The corresponding equations are given in (12) - (15) where we denote the joint GE parameters by α and β instead of α_{joint} and β_{joint}, respectively.

$$P_{0,0} = \Big((\pi_G(1-\alpha)^2 + \pi_B\beta(1-\alpha)) \times (1 - p^1_{disjoint}) \times (1 - p^2_{disjoint})\Big) \times Prob(j_{joint} + j^1_{disjoint} \le t^1_{max} \text{ and } j_{joint} + j^2_{disjoint} \le t^2_{max}) \quad (12)$$

$$\begin{aligned} P_{0,1} = &\Big((\pi_G(1-\alpha)\alpha + \pi_B\beta\alpha) \times (1 - p^1_{disjoint})\Big) \times Prob(j_{joint} + j^1_{disjoint} \le t^1_{max}) \\ &+ \Big((\pi_G(1-\alpha)^2 + \pi_B\beta(1-\alpha)) \times (1 - p^1_{disjoint}) \times p^2_{disjoint}\Big) \times Prob(j_{joint} + j^1_{disjoint} \le t^1_{max}) \\ &+ \Big((\pi_G(1-\alpha)^2 + \pi_B\beta(1-\alpha)) \times (1 - p^1_{disjoint}) \times (1 - p^2_{disjoint})\Big) \times Prob(j_{joint} + j^1_{disjoint} \le t^1_{max} \text{ and } j_{joint} + j^2_{disjoint} > t^2_{max}) \end{aligned} \quad (13)$$

$$\begin{aligned} P_{1,0} = &\Big((\pi_G\alpha\beta + \pi_B(1-\beta)\beta) \times (1 - p^2_{disjoint})\Big) \times Prob(j_{joint} + j^2_{disjoint} \le t^2_{max}) \\ &+ \Big((\pi_G(1-\alpha)^2 + \pi_B\beta(1-\alpha)) \times p^1_{disjoint} \times (1 - p^2_{disjoint})\Big) \times Prob(j_{joint} + j^2_{disjoint} \le t^2_{max}) \\ &+ \Big((\pi_G(1-\alpha)^2 + \pi_B\beta(1-\alpha)) \times (1 - p^1_{disjoint}) \times (1 - p^2_{disjoint})\Big) \times Prob(j_{joint} + j^1_{disjoint} > t^1_{max} \text{ and } j_{joint} + j^2_{disjoint} \le t^2_{max}) \end{aligned} \quad (14)$$

$$P_{1,1} = 1 - P_{0,0} - P_{0,1} - P_{1,0} \quad (15)$$

V. Simulation Results

We have conducted several simulations to demonstrate the efficacy of the proposed multi-path selection method for a real-time streaming application. To this end, we generated a random Internet topology and by varying the topology parameters we compared different streaming methods in terms of video quality.

We used two standard test sequences Table Tennis and Flower Garden in our simulations to stream from a server to a client with a delay tolerance of 200 ms. Each sequence comprised of 150 frames. The resolutions were 352 x 240 pixels and the frame rate was 30 frames per second. To make quantitative comparisons, we used the PSNR measure given by $\text{PSNR} = 10 \times \log_{10}(\frac{255^2}{MSE})$, where MSE stands for the mean squared error. Among various multiple description video coders, we preferred the time-domain partitioning method with two descriptions.

In the simulations, the single description (SD) and multiple description (MD) encoded streams were produced with a standard MPEG-2 encoder at the available rates. Note that we were not interested in any particular video encoding scheme or concealment technique that could be applied to increase the streaming quality. Here, we performed a simple concealment technique that repeated the information from the last available frame in case of a bursty loss. For the isolated single errors, we concealed the missing frame with its neighboring frames (the other description in the MD case). Naturally, more intelligent encoding and concealment techniques would result in higher streaming qualities for both SD and MD encoded streams.

A. Internet Topology Simulations

In order to generate a random Internet topology, we used GT-ITM [12]. The resulting topology consisted of 192 nodes with 4 mesh-connected transit domains, 8 nodes per transit domain, 1 stub per transit node and 5 nodes in a stub domain.[2] An overlay node (O-node) was associated with each transit domain node to enable multi-path routing. Notice that

[2]Since we were interested in streaming from a stub domain to another one (where the server and client were located, respectively), we kept the number of stub domains per transit domain node small.

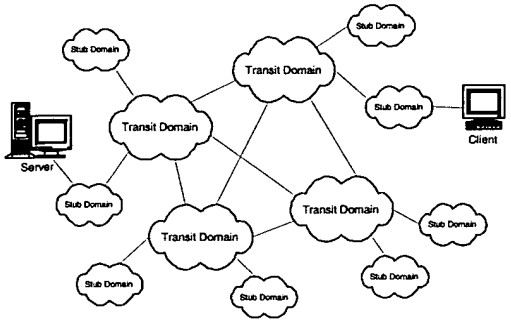

Fig. 4. An Internet topology with 4 transit domains.

multi-path routing capability was intended within only transit domains, not stub domains. An illustrative topology with 4 transit domains is shown in Fig. 4.

For the link capacities, we assigned 1.0 Gbps to transit-to-transit edges; 100 Mbps to transit-to-stub edges; 5 Mbps and 2 Mbps to stub-to-stub edges. The available bandwidth on each link ($b_{u,v}$) was chosen randomly in the range of 20%-80% of its capacity. The packet loss probabilities ($p_{u,v}$) were randomly assigned proportional to the link utilizations in the range of 0.1% - 1.0% for all links except one link in each domain (which was also randomly chosen to emulate congestion), whose loss rate was between 10%-20%. On the other hand, we assigned delays ($t_{u,v}$) in the range of 10 - 30 ms for the transit-domain links and 5 - 10 ms for the stub-domain links. Lastly, average jitter values for all links ($j_{u,v}$) were randomly chosen a value between 0 - 10 ms. These averages were then used to introduce exponentially distributed jitter for each packet.

On this topology, we randomly chose a server-client pair such that the server and client were separated by at least two transit domains. We compared three different streaming methods namely, the shortest path, maximally link-disjoint path and optimal multi-path streaming. For the shortest path, we chose the one with minimum number of hops and if this metric was equal for two or more paths, we selected the one with the minimum delay. On the other hand, for the maximally link-disjoint path pair, we first generated and evaluated all possible path pairs. Then, we looked for the pairs with minimum number of overlapping links to minimize the statistical dependency between the paths and then chose the pair with minimum total end-to-end delay. Notice that in doing so, we not only approximated the totally link-disjoint pairs, which were supposed to exist in [3], [4], but also selected shorter ones to avoid quality degradation due to the delayed packets. Lastly, for optimal multi-path streaming, based on the network statistics we selected the pair among all possible ones that minimized (4). While SD encoded video was used in the first method, we used the MD encoded video in the second and third methods.

We ran 10 different simulations for each test sequence by choosing different congested links at each run.[3] This analysis allowed us to observe the variations in the performances of three streaming methods. In the last simulation (Run #10) for the TABLE TENNIS sequence, we deliberately congested the link by which the stub domain including the server was connected to its transit domain. In this case, as expected, all three streaming methods performed poor since all streams had no choice other than to be routed over the same congested link. Obviously, path diversity becomes less effective when the congestion occurs on the joint links before the multiple paths split. The results for the two test sequences are tabulated in Tables I and II, respectively. In each row, we present the average PSNR of the streamed video and total number of lost and delayed (obsolete) packets for a different run. Evidently, optimal multi-path streaming outperforms shortest path and maximally link-disjoint path streaming in all of these independent runs. There are prominent conclusions that can be drawn from these results. We elaborate on them next.

TABLE I

SIMULATION RESULTS FOR THE TABLE TENNIS SEQUENCE

	Shortest Path		Max. Link-Dis. Path		Optimal Multi-Path	
Run #	PSNR	Lost/Del.	PSNR	Lost/Del.	PSNR	Lost/Del.
1	33.1	4 / 1	33.2	4 / 2	33.5	3 / 2
2	25.7	28 / 3	30.2	15 / 3	33.0	6 / 5
3	32.5	6 / 1	33.2	5 / 2	33.2	3 / 4
4	32.4	7 / 1	33.3	5 / 1	33.5	3 / 2
5	25.7	28 / 3	30.1	18 / 3	33.2	3 / 4
6	29.3	18 / 2	30.1	19 / 4	33.0	5 / 5
7	31.6	8 / 2	33.5	4 / 1	33.5	4 / 1
8	25.2	32 / 3	30.2	17 / 2	33.3	3 / 3
9	32.4	6 / 2	32.0	10 / 3	33.3	3 / 3
10	25.8	28 / 2	27.2	25 / 4	27.3	24 / 5
Avg.	29.4	16.5 / 2.0	31.3	12.2 / 2.5	32.7	5.7 / 3.4

TABLE II

SIMULATION RESULTS FOR THE FLOWER GARDEN SEQUENCE

	Shortest Path		Max. Link-Dis. Path		Optimal Multi-Path	
Run #	PSNR	Lost/Del.	PSNR	Lost/Del.	PSNR	Lost/Del.
1	18.8	30 / 3	21.9	16 / 2	24.1	4 / 3
2	19.1	27 / 3	21.7	17 / 3	24.1	3 / 4
3	18.9	29 / 2	21.6	18 / 2	23.9	4 / 4
4	24.0	5 / 1	24.1	5 / 2	24.3	3 / 3
5	23.6	6 / 2	24.4	4 / 2	24.4	3 / 3
6	18.9	28 / 3	21.8	15 / 3	23.8	6 / 5
7	23.8	6 / 2	23.9	8 / 3	24.4	3 / 3
8	23.9	6 / 1	24.3	5 / 1	24.5	4 / 2
9	20.3	20 / 2	21.0	20 / 4	23.8	5 / 4
10	23.6	6 / 2	23.7	5 / 3	23.8	4 / 4
Avg.	21.5	16.3 / 2.1	22.9	11.3 / 2.5	24.1	3.9 / 3.5

B. Discussion

When we carefully examine the results, we observe that the shortest path streaming often degrades the quality to an unacceptable level. This occurs when a link on the shortest path is heavily congested. Naturally, the maximally link-disjoint path streaming is able to reduce the adverse effect of the congested link by exploiting the path diversity and improve the average quality over the shortest path streaming. However,

[3]We repeated the simulations 100 times to obtain statistically reliable results. Also, we chose an average burst length, L, of 4.

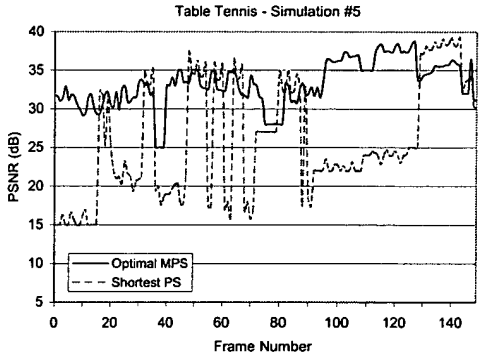

Fig. 5. The PSNR comparisons of the shortest path and optimal multi-path streaming methods for TABLE TENNIS.

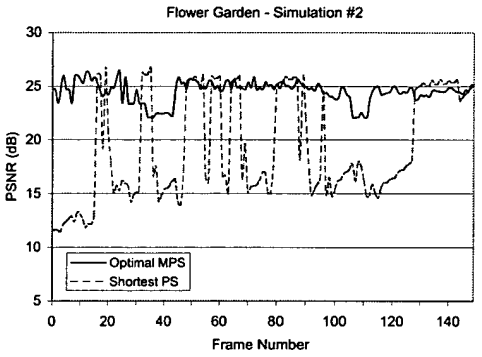

Fig. 6. The PSNR comparisons of the shortest path and optimal multi-path streaming methods for FLOWER GARDEN.

its performance is still below the optimal multi-path streaming. Hence, we conclude that minimizing the dependency between the paths without considering their characteristics does not necessarily result in the maximal streaming quality.

Another interesting point is the distribution of individual frame qualities. The plots in Fig. 5 and Fig. 6 show the PSNR values of each frame for the videos streamed over the shortest path and optimal path pair. These plots are extracted from the 5^{th} and 2^{nd} simulations for the TABLE TENNIS and FLOWER GARDEN sequences, respectively. As seen from the plots, the poor video quality persists for the SD encoded stream during a bursty loss. Moreover, the wide fluctuations further degrades the perceptual quality. However, the video quality slightly deviates from its average value for the MD encoded stream unless both descriptions are lost simultaneously. This feature provides smoother streaming experience to the client.

The results also show that optimal multi-path streaming experienced a slightly higher number of delayed packets compared to other methods. This is mainly because of the fact that in this method, a longer but less congested link may be preferred over a shorter but more congested link as long as the delay tolerance allows to do so. On the other hand, when the shortest path (the maximally link-disjoint paths) is not congested, *i.e.*, it is a *good* path, the proposed method may also choose this shortest path (the maximally link-disjoint paths) as well. Because of this flexibility, optimal multi-path streaming performs better than its opponents. Lastly, we observed the tendency of multi-path routing in load balancing. Rather than overloading the shortest path, the total load is shared between different paths.

VI. CONCLUSIONS

In this paper, we modelled multi-path streaming and proposed an optimal multi-path selection method for MD streaming. This method yields the maximal video quality at the client by evaluating the network conditions and application requirements. The simulations show that the optimal multi-path selection attains considerably large quality improvements over both the shortest and maximally link-disjoint path streaming. At the implementation side, we achieve the MD streaming by the use of overlay infrastructure. In doing so, we do not advocate any particular congestion or routing mechanism in the underlying network. However, additional improvement can be achieved if the network nodes become multi-path routing aware, *e.g.*, one of the descriptions may be prioritized over the others to prevent concurrent description losses.

In this paper, we paid attention to the two-path case due to its tractability and practicability, however it is straightforward to generalize the proposed model to more than two paths at the expense of increased computational complexity. Also, the framework can be adapted to other streaming media such as audio and 3-D graphics.

REFERENCES

[1] J. G. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in *Visual Communications and Image Processing (VCIP)*, 2001.

[2] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee, "On multiple description streaming with content delivery networks," in *IEEE INFOCOM*, 2002.

[3] Y. J. Liang, E. Setton, and B. Girod, "Channel-adaptive video streaming using packet path diversity and rate-distortion optimized reference picture selection," in *IEEE Fifth Workshop on Multimedia Signal Processing*, 2002.

[4] S. Lin, S. Mao, Y. Wang, and S. Panwar, "A reference picture selection scheme for video transmission over ad-hoc networks using multiple paths," in *IEEE ICME*, 2001.

[5] R. Chow, C. Lee, and J. C. Liu, "Traffic dispersion strategies for multimedia streaming," in *IEEE Eighth Workshop on Future Trends of Distributed Computing Systems*, 2001.

[6] J. G. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2000.

[7] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Processing Mag.*, vol. 18, no. 5, pp. 74–93, 2001.

[8] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient overlay networks," in *18th ACM Symposium on Operating Systems Principles*, 2001.

[9] J. Touch and S. Hotz, "The X-Bone," in *Third Global Internet Mini-Conference at IEEE GLOBECOM*, 1998.

[10] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," in *IEEE INFOCOM*, 1999.

[11] J. Wenyu and H. Schulzrinne, "Modeling of packet loss and delay and their effects on real-time multimedia service quality," in *NOSSDAV*, 2000.

[12] E. W. Zegura, K. L. Calvert, and M. J. Donahoo, "A quantitative comparison of graph-based models for Internet topology," *IEEE/ACM Trans. Networking*, vol. 5, no. 6, pp. 770–783, 1997.